

NAME

`slurm_free_front_end_info_msg`, `slurm_load_front_end`, `slurm_print_front_end_info_msg`, `slurm_print_front_end_table`, `slurm_sprint_front_end_table` – Slurm front end node information reporting functions

SYNTAX

```
#include <stdio.h>
#include <slurm/slurm.h>

void slurm_free_front_end_info_msg (
    front_end_info_msg_t *front_end_info_msg_ptr
);

int slurm_load_front_end (
    time_t update_time,
    front_end_info_msg_t **front_end_info_msg_pptr,
);

void slurm_print_front_end_info_msg (
    FILE *out_file,
    front_end_info_msg_t *front_end_info_msg_ptr,
    int one_liner
);

void slurm_print_front_end_table (
    FILE *out_file,
    front_end_info_t *front_end_ptr,
    int one_liner
);

char *slurm_sprint_front_end_table (
    front_end_info_t *front_end_ptr,
    int one_liner
);
```

ARGUMENTS

front_end_info_msg_ptr

Specifies the pointer to the structure created by **slurm_load_front_end**.

front_end_info_msg_pptr

Specifies the double pointer to the structure to be created and filled with the time of the last front end node update, a record count, and detailed information about each front_end node. Detailed front_end node information is written to fixed sized records and includes: name, state, etc. See `slurm.h` for full details on the data structure's contents.

front_end_ptr

Specifies a pointer to a single front end node record from the *front_end_info_msg_ptr* data structure.

one_liner

Print one record per line if non-zero.

out_file Specifies the file to print data to.

update_time

For all of the following informational calls, if *update_time* is equal to or greater than the last time changes were made to that information, new information is not returned. Otherwise all the configuration, job, node, or partition records are returned.

DESCRIPTION

slurm_free_front_end_info_msg Release the storage generated by the **slurm_load_front_end** function.

slurm_load_front_end Returns a `front_end_info_msg_t` that contains an update time, record count, and



array of records for all front end nodes.

slurm_print_front_end_info_msg Prints the contents of the data structure describing all front end node records from the data loaded by the **slurm_load_front_end** function.

slurm_print_front_end_table Prints to a file the contents of the data structure describing a single front end node record loaded by the **slurm_load_front_end** function.

slurm_psrprint_front_end_table Prints to memory the contents of the data structure describing a single front end node record loaded by the **slurm_load_front_end** function.

RETURN VALUE

On success, zero is returned. On error, -1 is returned, and Slurm error code is set appropriately.

ERRORS

SLURM_NO_CHANGE_IN_DATA Data has not changed since **update_time**.

SLURM_PROTOCOL_VERSION_ERROR Protocol version has changed, re-link your code.

SLURM_PROTOCOL_SOCKET_IMPL_TIMEOUT Timeout in communicating with Slurm controller.

EXAMPLE

```
#include <stdio.h>
#include <slurm/slurm.h>
#include <slurm/slurm_errno.h>

int main (int argc, char *argv[])
{
    int i;
    front_end_info_msg_t *front_end_info_ptr = NULL;
    front_end_info_t *front_end_ptr;

    /* get and dump some node information */
    if ( slurm_load_front_end ((time_t) NULL,
        &front_end_buffer_ptr) ) {
        slurm_perror ("slurm_load_front_end error");
        exit (1);
    }

    /* The easy way to print... */
    slurm_print_front_end_info_msg (stdout, front_end_buffer_ptr, 0);

    /* A harder way.. */
    for (i = 0; i < front_end_buffer_ptr->record_count; i++) {
        front_end_ptr = &front_end_buffer_ptr->front_end_array[i];
        slurm_print_front_end_table(stdout, front_end_ptr, 0);
    }

    /* The hardest way. */
    for (i = 0; i < front_end_buffer_ptr->front_end_count; i++) {
        printf ("FrontEndName=%s StateCode=%u\n",
            front_end_buffer_ptr->front_end_array[i].name,
            front_end_buffer_ptr->front_end_array[i].node_state);
    }
    slurm_free_front_end_info_msg (front_end_buffer_ptr);
    exit (0);
}
```

NOTES

These functions are included in the libslurm library, which must be linked to your process for use (e.g. "cc -lslurm myprog.c").

Some data structures contain index values to cross-reference each other. If the *show_flags* argument is



not set to `SHOW_ALL` when getting this data, these index values will be invalid.

COPYING

Copyright (C) 2010 Lawrence Livermore National Security. Produced at Lawrence Livermore National Laboratory (cf, `DISCLAIMER`). `CODE-OCEC-09-009`. All rights reserved.

This file is part of Slurm, a resource management program. For details, see [<https://slurm.schedmd.com/>](https://slurm.schedmd.com/).

Slurm is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Slurm is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

SEE ALSO

`scontrol(1)`, `slurm_get_errno(3)`, `slurm_load_node(3)`, `slurm_perror(3)`, `slurm_strerror(3)`

